Cluster Analysis of Stocks

Project Proposal Paper

Jordan Zane Caleb Clough Matt Dyer Evan Linden

PROBLEM STATEMENT

This project will aim to mine extensive longitudinal American stock/etf market data to identify interesting trends.

In particular, our group aims to cluster individual stocks into groups to see what individual stocks tend to "move" together. Ideally this analysis will uncover greater nuance and cross relationships outside of typical stock groupings (technology, manufacturing, finance, etc.) and allow for novel (or at least modified) classification grouping schemes.

Potential applications of the results of this project could be to more accurately predict stock movements, or at least correlations between certain stocks/etf that are found to be in similar groups. Not only could this data be useful for identifying undervalued assets, but it could also be useful to hedge risk in a portfolio by ensuring multiple cluster groups are represented in a portfolio-- ensuring groups tend to move independently, while having multiple groups present in a portfolio can ensure the overall portfolio balance does not swing wildly.

DATA SETS

Our main dataset is a .csv containing 5 years of AMEX, NYSE, and NASDAQ end of day data. The source of this data is Kaggle and the URL is as follows: https://www.kaggle.com/qks1lver/amex-nyse-nasdaq-stock-histories?select=fh_5yrs.csv

The overall size of the dataset is 712 MB, with a total of 6852039 data points. The data is structured such that each stock ticker represented has (up to) 5 years of daily data.

Each data point has a total of 8 attributes. They are as follows, with information about the type of variable each is:

- Date - Ordinal
- Volume - Numeric (Ratio)
- Open - Numeric (Ratio)
- High - Numeric (Ratio)
- Low - Numeric (Ratio)
- Close - Numeric (Ratio)
- Adj Close - Numeric (Ratio)
- Symbol - Categorical

The data appears to be very clean, but likely some minor cleaning will need to be done, as described in the aforementioned sections.

Additionally, we have a secondary data set to supplement the main dataset in the event that the primary dataset is not sufficient to meet our project objectives. This secondary dataset can be

found at: https://www.kaggle.com/borismarjanovic/price-volume-data-for-all-us-stocks-etfs

This dataset is structured a bit differently than the primary dataset. Each stock and etf has a separate txt file with longitudinal data. Each txt file is structured with the data being comma
delimited, and thus can be treated as a csv with minimal modification. In total there are 8539 files totaling 771 MB. Each file contains a total of 7 attributes. They are as follows, with information about the type of variable each is:

- Date - Ordinal
- Open - Numeric (Ratio)
- High - Numeric (Ratio)
- Low - Numeric (Ratio)
- Close - Numeric (Ratio)
- Volume - Numeric (Ratio)
- Open Int - Numeric (Ratio)

This data also appears to be very clean, but may require substantial post-processing to integrate with the primary dataset in the event that it is needed

LITERATURE SURVEY

Studies come from Kaggle:

- Which stocks should I invest in? - included daily percentage change in price and volume, Correlation of percent change, Predicting expected gain and loss for each ticker, and growth trends of tickers
- Stock diversity analysis - analysing clusters to identify stocks with opposite trends
- Linear regression of stock histories - using a regression model to predict stock prices
- Wsb stock analysis - using popular tickers mentioned on reddit.com to analyze growth

PROPOSED WORK

1 Data Preprocessing

Once we have collected our complete data set from Kaggle, we will need to clean the data before we do any analysis with it. The data cleaning process will begin by dropping any NULL values from the dataset. Then the next step is to replace any string values such as '$' with a ' ', so we do not have any strings in our data set. Considering we are going to be comparing the stocks based on their prices over time, we will also need to convert the dates using the Pandas function "to_datetime", so that Python recognizes them as Dates, not strings. The last step of data cleansing is to locate and remove any outliers from the data set. Once these steps have been completed, the dataset will have been cleansed of any imperfections and is now primed for any data-mining we need to pursue.

For data integration, we will be using potentially multiple stock datasets containing similar attributes. To avoid redundancy in the data, the standard stock symbols will be used for entity identification. The dates will be converted to datetime using the Pandas library in Python to ensure that all dates are in a standard tractable format. This will ensure that no duplicate dates will be used for the same stock.

2 Data Integration

Our data analysis will rely on hierarchical clustering which requires high space and time complexity. Therefore, we will need to use considerable data reduction to ensure efficient and speedy mining techniques. Since we are primarily interested in the daily stock movement, we can reduce the dimensionality of the data by only using the open and close attributes to track daily stock movement. This will also allow us to avoid possible strong correlation in attributes such as close and adjusted close. In case the data needs to be further reduced for effective hierarchical clustering, we can turn to numerosity reduction techniques like sampling. For our time series analysis, we will reduce each cluster to an ARIMA regression model to forecast future stock movement.

3 Literature Survey Comparison

First off there is only one piece of work that is related to analysis using clusters, the rest are analysing the data set using methods such as correlation of percentage change and linear regression. The one piece of prior work, "Stock Diversity Analysis", analyzes with percentage change of day-to-day price changes which will be similar to ours, but what we will be doing differently is using hierarchical clustering. Using the ARIMA regression model we will hopefully be able to get more accurate and better predictions

DATA ANALYSIS AND TOOLS

1 Methods

First we will measure daily stock movement by subtracting the open price from the closing price of each stock for every day. Next, we will randomly select 80% of the days to be our training set for the model with the remaining 20% serving as our test set. We will then use agglomerative hierarchical clustering to determine similar groups of stocks according to their daily price movements.

This will be performed using the AgglomerativeClustering library in Python. Euclidean distance will be used as the similarity measure and mean distance will be the metric to determine the distance between clusters. These metrics are purely arbitrary and we may experiment with different distance metrics to assess the differences in clustering. Using a dendrogram to represent the process, we will select a termination condition that agrees with our domain knowledge. For example, if one cluster contains mainly tech stocks and energy sector stocks, we would continue the partitions until these groups of stocks are largely segregated. This will be a difficult judgment and require considerable domain research to determine when to stop clustering. After all, our goal is to find interesting and less obvious groups. Therefore, we want to avoid overfitting the data.

Finally, we will predict future movement of the means of these clusters using time series analysis in R. In particular, we will use the Hyndman-Khandakar algorithm. Using plots of the ACF of the residuals in conjunction with portmanteau residual tests, we will adjust the model accordingly until the residuals appear as white noise. Then, we will use the resulting models for each cluster to forecast future stock movement.

2 Progress Update

A major issue so far has been the concept validity of our project. The main objective is to identify similar stocks based on their movement. However, the measure of movement we choose can potentially affect the clusters. For example, Stock A could rise steadily throughout a given week. Meanwhile, Stock B could fall steadily throughout a given week with a large rise at the end. The daily movement of each stock would be different but the weekly measurement would be similar.

In addition, the magnitude of the stock movement can vary wildly. Stock A could increase slightly while Stock B could increase drastically. These two stocks could have different net gains compared to percent gains as well. With so many different ways to measure stock movement, it has proven challenging picking the right method to measure similar stocks.

Ultimately, the most practical conclusion that could be made from our data is which stocks move in tandem generally. The magnitude of the stock movement is not as important in this regard. The most important result is whether to long or short a cluster of stocks and to diversify our portfolio among varied clusters. Therefore, we decided to measure stock movement through a simple binary measure. If the stock moves up, we will assign that stock and day a 1. If the stock stays the same or decreases, we will assign that stock and day a 0. Using Euclidean distance, we will use agglomerative clustering to identify similar stock groupings.

Through this metric, we are essentially measuring stock similarity based on the direction of their movement each day. It is irrelevant how far the stock price moves. Thus, there are some limitations in our interpretation. We are identifying stocks that move in the same direction but they may move at completely different magnitudes. Our interpretation of ‘similar’ only goes as far as stocks expected to move in tandem.

EVALUATION METHODS AND TOOLS

For our purposes, we are assuming the ground truth is essentially unavailable because we are looking for clusters of seemingly disconnected stocks. Therefore, we will rely on intrinsic methods to evaluate how well the clusters are separated and how compact the clusters are.

For a general guideline, we will use the elbow method to determine if the optimal number of clusters were used. However, we will still ultimately rely on our own judgment and domain expertise to decide on the final number of clusters. We will ensure that our judgment generally agrees with the elbow method.

Our primary evaluation method will be through cross-validation. We will test our models that use a different number of clusters, different distance measures, etc. and evaluate the sum of the squared differences in between the points in a cluster and their centroid as well as the squared differences with the other centroids. A smaller intradistance and a larger interdistance signifies a better fitting model.

MILESTONES

1. Locate Data Set(s) - Already Done
2. Data Preprocessing - 7/12-7/18
3. Data Analysis - 7/19-7/25
    a. Daily Stock Movement
    b. Select Training and Test Sets
    c. Clustering code
    d. Analyzing data
4. Making a conclusion from our analysis - 7/27-8/1

1 Milestones Completed

- Locate Data Set(s)
- Data Preprocessing
- Data Analysis
    - Daily Stock Movement
    - Select Training and Test Sets

2 Milestones Todo / In Progress

- Data Analysis
    - Clustering Code
    - Analyzing Data
- Making final conclusion from our analysis

RESULTS SO FAR

As previously mentioned, the main goal of our current efforts has been to preprocess our main data csv such that a clustering analysis can be performed effectively. In order to simplify the data to allow for easier training, the group decided to identify instances where a certain stock has increased day-over-day, with a simple 1 (yes) or 0 (no) scheme. This will allow us to then do a market-basket type analysis and/or allow distance calculations between stocks for a given sequence of days to find stocks or groups of stocks that tend to move together-- the stated goal for this analysis.

As such, a derived column for the data was produced with the following criteria:

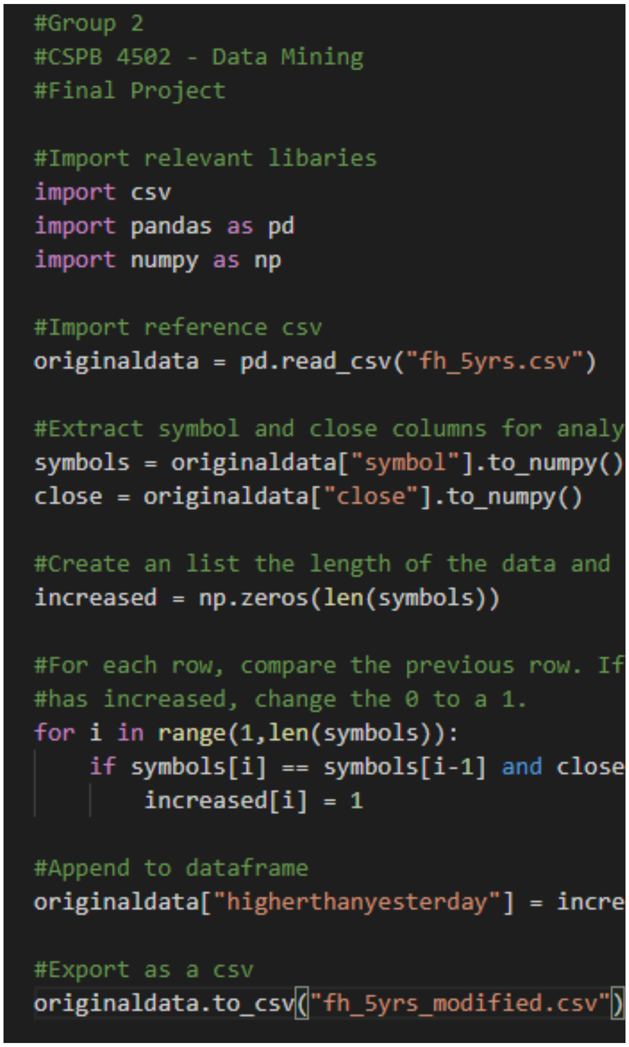
1. IF a row of data has the same ticker symbol as the row before it
    a. To ensure comparisons only occur between the same stock

2. AND the row of data's value for close is greater than the row before it

. To enforce the rule that only days where the value has been increased DoD shall get a "1"

3. THEN the row shall get a 1 in a new column called "higherthanyesterday"

In order to accomplish this, the group built a python script utilizing Pandas and NumPy to do the associated csv extractions, calculations, and repackaging. The source code for the script was as follows:

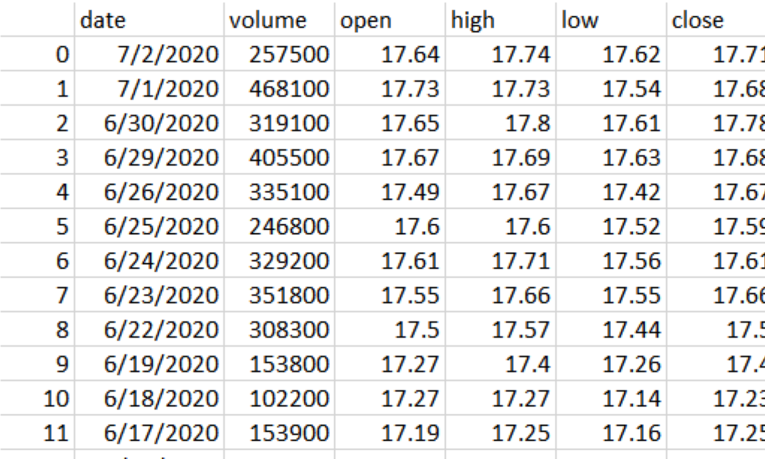


```
#Group 2
#CSPB 4502 - Data Mining
#Final Project

#Import relevant libaries
import csv
import pandas as pd
import numpy as np

#Import reference csv
originaldata = pd.read_csv("fh_5yrs.csv")

#Extract symbol and close columns for analy
symbols = originaldata["symbol"].to_numpy()
close = originaldata["close"].to_numpy()

#Create an list the length of the data and
increased = np.zeros(len(symbols))

#For each row, compare the previous row. If
#has increased, change the 0 to a 1.
for i in range(1,len(symbols)):
    if symbols[i] == symbols[i-1] and close
        increased[i] = 1

#Append to dataframe
originaldata["higherthanyesterday"] = incre

#Export as a csv
originaldata.to_csv("fh_5yrs_modified.csv")
```

Figure 1. Source code for "higherthanyesterday" derived attribute.

Following this, the resulting csv was opened in Excel to confirm the addition of the derived attribute column. As well, this additional column underwent extensive spot checking of the values to confirm script performance. Following these checks, it was determined that the script performed correctly and thus the augmented dataset is now ready for further analysis as the project progresses.

See below for a reference screenshot of the augmented csv, showing the additional column and correct values:



	date	volume	open	high	low	close
0	7/2/2020	257500	17.64	17.74	17.62	17.71
1	7/1/2020	468100	17.73	17.73	17.54	17.68
2	6/30/2020	319100	17.65	17.8	17.61	17.78
3	6/29/2020	405500	17.67	17.69	17.63	17.68
4	6/26/2020	335100	17.49	17.67	17.42	17.67
5	6/25/2020	246800	17.6	17.6	17.52	17.59
6	6/24/2020	329200	17.61	17.71	17.56	17.61
7	6/23/2020	351800	17.55	17.66	17.55	17.66
8	6/22/2020	308300	17.5	17.57	17.44	17.5
9	6/19/2020	153800	17.27	17.4	17.26	17.4
10	6/18/2020	102200	17.27	17.27	17.14	17.23
11	6/17/2020	153900	17.19	17.25	17.16	17.25

Figure 2. Snip of augmented csv opened in Excel.

A link to our updated data is as follows:

https://drive.google.com/drive/folders/17RyZ9pbuVMTZXn2SvbAhO8fvzA-08Kbv?usp=sharing